Movie Lens

Piri Yebra

2023-02-02

Executive summary

The purpose of this project is to make a movie recommendation system using several tools that have been taught throughout the course series. We were given train and test datasets with user, movie, genre, and rating information. With R code, I developed calculations in the train set and applied them in the test set.

My first approaches were using the caret package with the knn (k nearest neighbor) algorithm. It soon became clear that this approach was not working because of the high volume of data being processed that resulted in memory errors.

I then changed my approach and used user and movie averages within the train set and then applied such averages to the test set. This approach executed ok in my laptop and could get a reasonable RMSE when evaluating my prediction vs the test set real rating.

Method

The first step was to calculate the movie rating mean without user or movie segmentation.

```r
mu <- mean(edx$rating)
mu
```

```
## [1] 3.512465
```

Next step was to calculate the movie rating averages, as a difference from the overall mean, without considering user ratings.

```r
movie_avgs <- edx %>%
  group_by(movieId) %>%
  summarize(b_i = mean(rating - mu))
```

Final step was to calculate user rating averages, as a difference from the overall mean and the movie rating mean.

```r
user_avgs <- edx %>%
  left_join(movie_avgs, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating - mu - b_i))
```

Results

With movie and user averages, predicted ratings were calculated given the test set.

```
predicted_ratings <- final_holdout_test %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  mutate(pred = mu + b_i + b_u )
```

The way to test the prediction against the real rating is by calculating RMSE

```
model_movie_user_rmse <- RMSE(predicted_ratings$pred, predicted_ratings$rating)
model_movie_user_rmse
```

```
## [1] 0.8653488
```

Conclusion

In this movie prediction problem, all the input given was historic ratings data. As other machine learning problems, there are no mathematical formulas to calculate ratings. Training data has to be used to make predictions. For this particular exercise, data was given in a clean dataset and split into training and test sets. Usually, data scientists have to perform these steps.

During the last course, powerful machine learning algorithms implemented in the R caret package were explained. But high data volume made difficult the use of this tool so I had to change the approach. I am challenged to learn more about how to handle high data volume and make use of such algorithms.

With the method used, I took into consideration user and movie ratings. While getting a 0.8653 RMSE, I reflect that the result can be better if genres would have been taken into consideration. There is still much to learn and apply using the concepts and techniques of these data science courses.

Another interesting challenge in real life problems is that new variables come into play. For example, in a movie recommendation system, new users need recommendations or new movies need an initial predicted rating before having any recommendation. Data science algorithms have to take all these variables into account.